2019

A Statistical Analysis and Machine Learning of Genomic Data

Jongyun Jung
Minnesota State University, Mankato

MINNESOTA STATE UNIVERSITY,
MANKATO

MASTER OF SCIENCE THESIS

A Statistical Analysis and Machine Learning of Genomic Data

Author:
Jongyun JUNG

Supervisor:
Dr. Mezbahur RAHMAN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Mathematics and Statistics

April, 2019

This thesis has been examined and approved by the following members of the student's committee.

Signed:

Date:

Advisor/Chair Person,
Dr. Mezbahur Rahman,
Professor of Statistics,
Minnesota State University, Mankato.

Signed:

Date:

Committee Member,
Dr. In-Jae Kim,
Professor of Mathematics,
Minnesota State University, Mankato.

Signed:

Date:

Committee Member,
Dr. Ruijun Zhao,
Professor of Mathematics,
Minnesota State University, Mankato.

Declaration of Authorship

I, Jongyun JUNG, declare that this thesis titled, "A Statistical Analysis and Machine Learning of Genomic Data" and the work presented in it are my own. I confirm that:

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Every being cries out silently to be read differently."

Simone Weil, Gravity and Grace

Abstract

Machine learning enables a computer to learn a relationship between two assumingly related types of information. One type of information could thus be used to predict any lack of informaion in the other using the learned relationship. During the last decades, it has become cheaper to collect biological information, which has resulted in increasingly large amounts of data.

Biological information such as DNA is currently analyzed by a variety of tools. Although machine learning has already been used in various projects, a flexible tool for analyzing generic biological challenges has not yet been made. The recent advancements in the DNA sequencing technologies (next-generation sequencing) decreased the time of sequencing a human genome from weeks to hours and the cost of sequencing a human genome from million dollars to a thousand dollars. Due to this drop in costs, a large amount of genomic data are produced.

This thesis implemented the supervised and unsupervised machine learning algorithms for the genomic data. Distances are an integral part of all machine learning algorithms and hence play a central role in the analysis of most genomic data. The distance that is used for any particular task can have a profound effect on the output of the machine learning method and thus, it is essential that users ensure that the same distance method is used when comparing machine learning algorithms.

Acknowledgements

This thesis is written as if it is to be read by a fellow master student. By reading the thesis, the reader should get some background information into some of the main topics and challenges within both machine learing and a statistical analysis of genomic data.

Personally, I had little or no prior knowledge about biology and DNA sequencing altogether. It has been both interesting and challenging to get to know and learn about these complex fields. The learning curve has been long and steep. I had never been able to do it without the help and support from my supervisors, family, colleagues and friends.

First of all, I would like to thank Prof. *Dr. Mezbahur Rahman* for giving me the *degree of freedom* to research the topics that I'm really interested in and giving me a direction while I was studying at MNSU. Also, I want to thank my commitee members, *Dr. In-Jae Kim* and *Dr. Ruijun Zhao* for sharing their knowledge and expertise.

Finally, I will thank my beautiful wife *Borami Kang* for her dedication, the loving support throughout the whole process and allowing me to have sometime to think about *myself* outside of $[\int, \sum, \mu, \lambda, \cdots]$ everyday. And I would like to dedicate this thesis to my son *Joshua Jiwoo Jung*.

Contents

List of Figures

This thesis is dedicated to My beloved Parents,
Jung-I Jung and Heesun Kang.

Chapter 1

Introduction

1.1 Motivation

Biological challenges are interesting because they deal with the very foundations of mankind. And also it is eventually connects with my ultimate research topic of *Precision Medicine*. Addressing the current challenges is key to develop medicine to both prevent and cure diseases. A Statistical and Machine learning is a tool, one among many, for addressing the challenges.

Machine learning aims to make computers learn models or patterns which could be used for analysis, interpretation and decision making. A computer may learn from mathematical techniques (i.e regression analysis), complex computer algorithms (data-mining, artificial intelligence), amongst others. Regression analysis (Altland, 1999; Friedman, Hastie, and Tibshirani, 2001) is a statistical technique for *understanding* and *interpreting* relationships between independent and dependent mathematical variables, by estimation of sample data. A (probable) relationship may be examined using various techniques which explains one or more depencent variables based on one or more independent variables using a statistical model. A model is deterministic if it explains (in a complete manner) the dependent variables based on the independent ones. In many real-world applications, this is not possible. Instead, statistical (or stochastic) models tries to approximate exact solutions, by evaluating probabilistic distributions. The decisions made by using such models may be supported by various indicators (e.g. a confidence interval). Creating models and using probability distributions and indicators for decision making and forecasting are closely related with machine learning, even though machine learning may be understood more widely since it also have a branch to artificial intelligence.

Personally, this thesis has given me the passion to explore and to expand in what ways, and possibly how well, machine learning could be used to answer current challenges or problems formulations by using available genomic data.

1.1.1 Related Work

Various projects (Larranaga et al., 2006; Plewczynski et al., 2006) has already applied various machine learning approaches to challenge dealing with biological (genomic) data. Much effort has resulted in machine learning techniques applicable for dealing with genomic data in the sense of reading, storing, learning and analyzing it. A common focus of such projects has mainly been towards creating, improving and optimizing one ore more models for a specific case. Thus, because the project goal is closely related to the machine learning goal of learning and prediction data with a highest possible accuracy.

1.1.2 Focus and Challenges

Challenges of building a generic and flexible machine learning application has been the following:

- The transformation and representation of genome (genomic) data in a way which enables standard machine learning algorithms to work on it.
- The adaptation of a tool implementation, within an already complex and existing framework.
- The adversity of building a tool which bridges the fields of machine learning and biology when not having dealth with any ot it previously.

1.1.3 Covered Topics

This thesis covered topics, which are devoted extra attention are:

- How to build a machine learning tool, having the flexibility and power to solve a wide range of both current and future biological challenges.
- Creating measures which capture genomic data and which are both flexible and reusable in multiple genomic data contexts.
- The treatment of the enormous available amount of data, when dealing with sparse data (rare cases) and skewness (imbalanced data).

1.1.4 Uncovered Topics

While this thesis deals with the already mentioned topics in the above, the work does not involve developing any new algorithms, nor adjusting or extending any existing ones. Though some results are presented, the focus is only to explain general concepts or uses of mentioned algorithms and techniques.

1.1.5 Method

Extensive research has been done searching through online resources and libraries to find material on (prior) work within the field of data mining, machine learning and bioinformatics. To get hands-on experience with machine learning applications, an online course in Genomic Data Science Specialization[1], has been completed while writing the thesis. The application development has throughout the thesis been implmented in '`Python`'[2] and '`RStudio`'[3].

[1] Genomic Data Science Specialization (https://www.coursera.org/specializations/genomic-data-science) course

[2] Python Website (https://www.python.org/)

[3] RStudio Website (https://www.rstudio.com/)

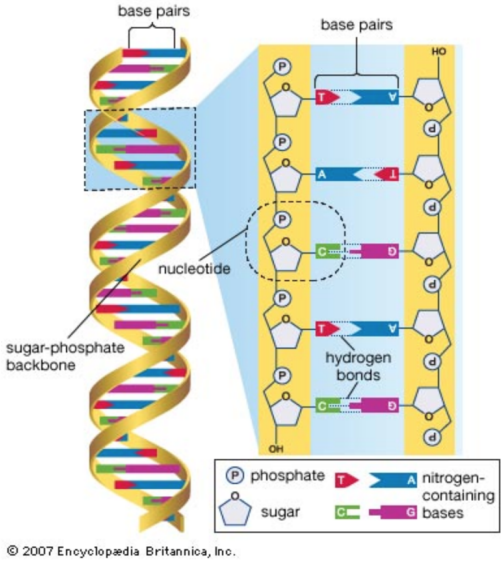

FIGURE 1.1: The DNA bases are adenine (A), thymine (T), guanine (G), and cytosine (C). Image taken from Encyclopedia Britannica.

1.2 Bioinformatics

In computer science, dealing with biological challenges is known as bioinformatics. Biologists and computer-scientists work together, using computer power, to gain insights of how the human body operates (internally). The insights might be used to create even better medicine to cure or prevent diseases. A key challenge has been to figure out what normal DNA looks like. Having proper understanding of what normal DNA is, facilitates detection of anomalies and changes.

1.2.1 DNA and the human genome

The human genome consists of about 3 billion base-pairs, known through the language of DNA (deoxyribonucleic acid) which consists of 4 bases, namely A, C, T and G as shown in Figure 1.1. The genome contains all our genes. More precisely, it contains the alleles which codes for the genes, but may differ on base-pairs, due to changes such as mutations.

1.2.2 DNA sequencing

DNA sequencing (Kircher, 2012) is the process of *reading* biological material and translating it into a computer readable data representation which may be used by scientists and researchers for a multiple of analytical purposes. The sequencing process is complex and introduces many challenges such as gaps between reads, lack of coverage and various other sequencing errors.

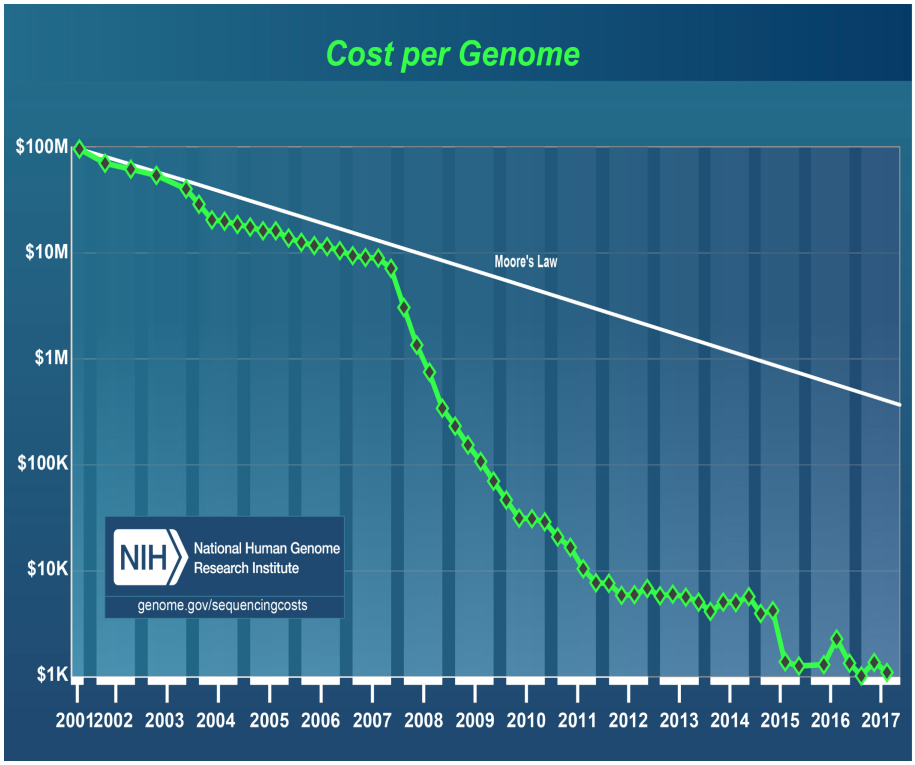

FIGURE 1.2: The cost of genome sequencing over the last 17 years (Wetterstrand, 2013)

1.2.3 DNA Sequencing Technologies

The DNA Sequencing procedure attempts to determine the exact arrangements of the nucleotides (*adenine, guanine, cytosine* and *thymine*) inside a DNA molecule. A wide range of different sciences including molecular biology, genetics, forensic studies and biotechology are benefiting the DNA sequencing technologies. (França, Carrilho, and Kist, 2002)

The advantage of the DNA sequencing technologies over the last 17 years has lessened the cost of sequencing a genome. As Figure 1.2, depicts the cost of sequencing a genome over the last 17 years. As seen, the figure isustrates Moore's Law as well. Moore's law assumes that the number of transitors, such as the computation power, is going to be doubled every two years. (Moore et al., 1965) Keeping up with Moore's law is considered to be remarkable successful in technological advancements. As Figure 1.2 shows, the DNA sequencing technologies had been keeping up with Moore's law until 2007. In 2005, the Next Generation Sequencing (NGS) technologies are introduced ((Schuster, 2007)) and consequently, the DNA sequencing technologies started to improve beyond Moore's law. By the advent of the next-generation sequencing, the cost of sequencing a genome is dropped to a mere thousand dollars from millions of dollars.

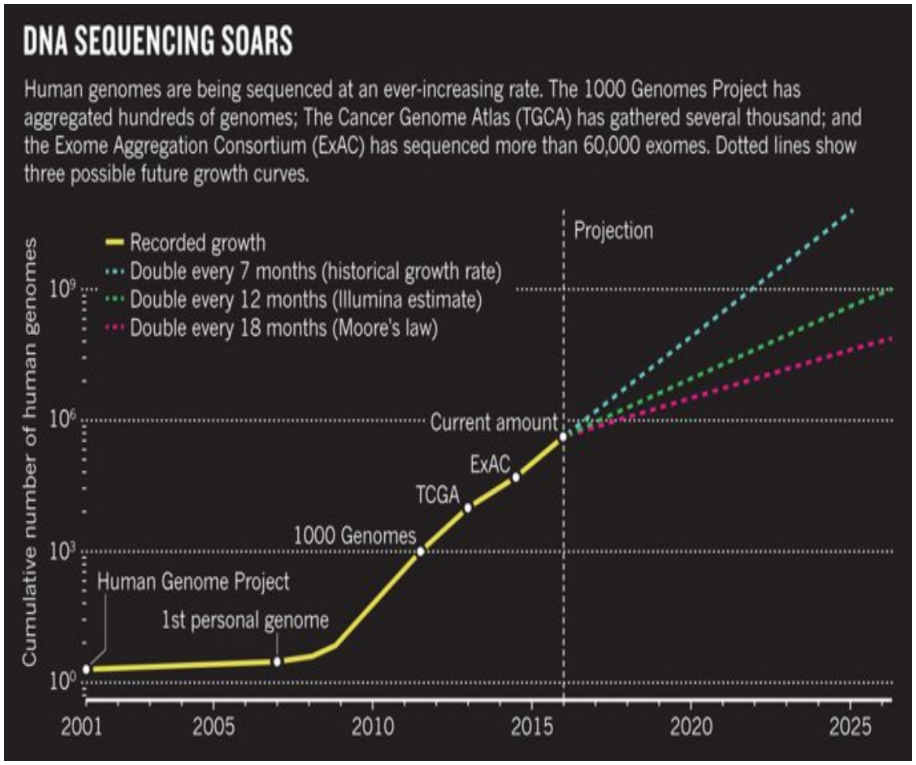

FIGURE 1.3: The number of sequenced human genomes over theThe number of sequenced human genomes over the years (Eisenstein, 2015)

1.2.4 Analysis of Genomic Data

As a result of the dramatic drop in the sequencing cost, the amount of sequenced genome data is significantly increasing. As Figure 1.3 shows the growth of the cumulative number of human genomes throughout the years. This amounts of available genomic data enabled the establishment of large scale sequencing data projects including The Cancer Genome Atlas (TCGA) (Tomczak, Czerwińska, and Wiznerowicz, 2015), The Encyclopedia of DNA Elements (ENCODE) (Harrow et al., 2012), and the 1000 Genomes Project Consortium (Consortium et al., 2012).

Those projects continuously collect and store sequencing data. In order to make an efficient use of the collected sequencing data, big data analysis techniques are essential.

1.2.5 Data formats

DNA sequencing information of a whole genome is often stored as a single file on a computer. The file may also contain meta-information like positions of chromosomes and genes, depending on the data-format specification. There are many formats, but the most popular are (Quinlan and Hall, 2010) and (Kent et al., 2010).

Chapter 2

Background

2.1 Machine Learning

Machine learning techniques (Alpaydin, 2009; Friedman, Hastie, and Tibshirani, 2001) has been increasingly popular over the last decades, due to the large amounts of available data and the access to freely available tools, e.g. Hadoop[1]. The field of machine learning offers many multi-purpose algorithms for operating on both small, large and huge datasets. In addition to this, many smart processing approaches has been proposed. Machine learning can also be viewed as extracting knowledge from data. However, the objective is not to store it, but to detect and use patterns for prediction purposes.

The key idea is to make a machine (computer) *learn* a model (hypothesis) by enough data of a given type, so it becomes able to identify one ore more patterns within it. Identified (learned) patterns may then be used for making estimates (predictions) on yet unseen data of similar type as the data which was used to learn the pattern. The amount of required data may vary based on the difficulty of the pattern to learn. The learning process is often referred to as *training*, while the process of making decisions is called *classification*.

There are mainly two types of learning. The first type, when data is given to the computer in addition to directly pointing out the pattern answer, is called *supervised learning*. The second type, when no such output are given, is called *unsupervised learning*. Sometimes, unsupervised learning is performed while providing answers at a later stage in the process to make adjustments or fine-tune one ore more parameters. This is called *semi-supervised learning*, since it is a combination of the two main types. Notice that other machine learning variants of the types do exists (e.g. reinforcement learning), but are not discussed in this thesis.

2.1.1 Supervised Learning

Supervised learning is to learn an hypothesis (model) using *answers* to help the machine figure out patterns. By this, the patterns to be learned is assumed to be known when the learning process begins. For the learning to have any meaning, there must be at least one pattern to learn. Thus, the outcome of

[1] http://hadoop.apache.org/

all instances (samples) could either represents the presence or absence of the pattern.

The supervision part, is (for each sample) to *tell* the machine if a pattern is present or not. A sample instance where a pattern is present is denoted a positive sample. Equally, a sample where a pattern is not present (absent) is denoted a negative sample.

Binary - and Multi-class Classification

A model (or hypothesis) which is used to predict two outcomes is knows as binary classification. For instance, it may predict or classify a sample to be either positive or negative. Multiclass classification is when there are more than two possible outcomes (classes). In general, a n- class classifier may classify n possible outcomes. There is no such thing as a one-class classifier ($n = 1$) since there is no classes to distingish between.

In some cases a binary classifier may be used as a multiclass classifier. This is known as a *one-vs-all* or *one-vs-rest* classifier. This idea is to build a collection $\mathbb{C}$ of n binary classifiers (c), one for each class, and then select the i-th classifier which estimates the highests probability for a given sample x.

$$\underset{x}{\operatorname{argmax}}\, c(x) = \{c(x_i) | \forall\, j\, \exists\, c(x_j) \leq c(x_i) \wedge c_i, c_j \in \mathbb{C}\}$$

2.1.2 Unsupervised Learning

Unsupervised learning encourages the computer to figure out patterns by itself and learn an hypothesis without explicitly pointing out any answers for it. Such learning is particularly good in discovering segments within a data set, often by exploring relationship between huge amounts of data (big data). Examples of such segmentation could be detecting customer groups for targeted marketing or discovering solar system relationships. Applications which applies techniques from this field are usually somewhat related to artificial intelligence.

2.2 Distance Measures

Both supervised and unsupervised machine learning techniques require selection of a measure of distance between, or similarity among, the objects to be classified or clustered. Different measures of distance or similarity will lead to different machine learning performance. The appropriateness of a distance measure will typically depend on the types of features being used in the learning process.

Genomic experiments generate large and complex multivariate data sets. Machine learning approaches are important techniques in genomic data for the purpose of identifying patterns in expression among genes and/or biological samples, and for predicting clinical or other outcomes using gene expression data.

Inherent in every machine learning approach is a notion of a distance or similarity the objects to be clustered or classified. In general, any distance measure can be used with any machine learning algorithm. The choice of distance measure is probably more important than the choice of machine learning algorithm, and some attention should be paid to the selection of an appropriate measure for each problem.

Certain supervised learning methods, such as K-nearest neighbor classifiers, also involve explicitly specifying a distance. Although the choice of distance may not be as transparent for other supervised approaches, observations are in fact assigned to classes on the basis of their distances from objects known to be in the classes. For example, linear discriminant analysis is based on the Mahalanobis distance (Penny, 1996) of the observations from the class means.

2.2.1 Distances

Distances, metrics, dissimilarities, and similarities are related concepts. We provide some general definitions and then consider specific classes of distance measures.

Definition 2.2.1
Any function d that satisfies the following five properties is termed as a metric*:*

1. ***non-negativity*** $d(\mathbf{x}, \mathbf{y}) \geq 0$
2. ***symmetry*** $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
3. ***identification mark*** $d(\mathbf{x}, \mathbf{x}) = 0$
4. ***definiteness*** $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$
5. ***triangle inequality*** $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$

A function that satisfied only properties $1 - 3$ is termed a *distance*. For many of the techniques we will consider, distances are sufficient. Hence, we

will generally refer to distances (which include metrics) and only mention metrics specifically when properties 4 and 5 are relevant.

A *similarity function* S is more loosely defined and satisfies the three following properties:

1. **non-negativity** $S(\mathbf{x}, \mathbf{y}) \geq 0$
2. **symmetry** $S(\mathbf{x}, \mathbf{y}) = S(\mathbf{y}, \mathbf{x})$
3. $S(\mathbf{x}, \mathbf{y})$ increases in a monotone fashion as objects $\mathbf{x}$ and $\mathbf{y}$ are more and more *similar*.

A *dissimilarity* function satisfies 1 and 2, but for 3, $S(\mathbf{x}, \mathbf{y})$ increases as objects $\mathbf{x}$ and $\mathbf{y}$ are more and more *dissimilar*. It is worth noting that there is, in fact, no need to require symmetry although some adjustments generally need to be made if the measures are not symmetric. The airplane flight time between two cities is an example of an asymmetric distance.

Many options are available in selection of a distance for machine learning tasks. Because there are many different types of data (e.g., ordinal, nominal, continuous) and approaches for analyzing these data, the literature on distances is quite broad. References that consider the application of distances in either clustering or classification include: (Duda, Hart, and Stork, 2012).

We are most interested with a situation where G features have been measured for I observations, or samples. There is substantial interest in applying some form of machine learning to both the samples (e.g., to identify patients with similar patterns of mRNA expression) and the features (e.g., to identify genes with similar patterns of expression).

We distinguish between two main classes of distance measures. Consider computing the distance between the expression profiles of two genes across I samples. In the first approach, we view the gene expression profiles as two I-vectors in some space and compute distances in a pairwise (within-sample) manner. In contrast, the second approach ignores the natural pairing of observations and instead, views the two gene expression profiles as two different samples generated from the underlying probability density functions for mRNA expression measures. In this case, distances between densities or distribution functions are relevant.

2.2.2 Distances between points

For m-vectors $\mathbf{x} = (x_1, \cdots, x_m)$ and $\mathbf{y} = (y_1, \cdots, y_m)$ consider distances of the form

$$d(\mathbf{x}, \mathbf{y}) = F[d_1(x_1, y_1), \cdots, d_m(x_m, y_m)] \tag{2.1}$$

where the d_k are themselves distances for each of the $k = 1, \cdots, m$ features. We refer to these functions as *pairwise distance functions*(Gentleman et al., 2006), as the pairing of observations within features is preserved. This representation is quite general: there is no need for the d_k to be the same. In

particular, features may be of different types (e.g., the data may consis of a maxture of continuous and binary features) and may be weighted differentially (e.g., weighted Euclidean distance).

Common metrics within this class include the Minkowski metric, with $z_k = d_k(x_k, y_k) = |x_k - y_k|$ and $F(z_1, \cdots, z_m) = (\sum_{k=1}^{m} z_k^{\lambda})^{\frac{1}{\lambda}}$. Special cases of the Minkowski metric considered in this thesis are the Manhattan and Euclidean metrics corresponding to $\lambda = 1$ and $\lambda = 2$, respectively.

Euclidean metric is defined as

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{2.2}$$

Manhattan metric is defined as

$$d_{man}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i| \tag{2.3}$$

Correlation-based distance measures have been widely used in the genomic data literature (Eisen et al., 1998). They include one minus the standard Pearson correlation coefficient and one minus an uncentered correlation coefficient considered by (Eisen et al., 1998).

Pearson sample correlation distance is defined as

$$d_{cor}(\mathbf{x}, \mathbf{y}) = 1 - r(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2 \sum_{i=1}^{m}(y_i - \bar{y})^2}} \tag{2.4}$$

Cosine correlation distance is defined as

$$d_{eisen}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{|\sum_{i=1}^{m} x_i y_i|}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}} \tag{2.5}$$

which is a special case of Pearson's correlation with $\bar{x}$ and $\bar{y}$ both replaced by zero.

Note that we have transformed the correlations by substracting them from one. This is done so that two vectors that are strongly positively correlated are regarded as close together. Using this transformation, data that exhibit a strong negative correlation will be far apart. In some cases, you might want to treat negative and positive correlation similarly, and that can be achieved by using the absolute value of the correlation. Correlation-based measures are in general invariant to location and scale transformations and tend to group together genes whose expression patterns are linearly related. While correlation-based distances have many nice properties, they tend to be adversely affected by outliers and then the non-parametric versions are preferre (Eisen et al., 1998).

We can also use *Mahalanobis distance*. Consider a situation where a pair of vectors, $\mathbf{x}$ and $\mathbf{y}$ are generated from some multivariate distribution with

mean vector μ and variance-covariance matrix $\sum$. Then the Mahalanobis distance between them is defined as

$$(\mathbf{x} - \mathbf{y})' \sum^{-1} (\mathbf{x} - \mathbf{y}) \tag{2.6}$$

When $\sum$ is unknown, it is generally replaced with the sample variance-covariance matrix. In general terms, the Mahalanobis distance reflects the notion that the data are more variable in some directions than in others.

2.2.3 Distances between distributions

The distances enumerated in the preceding section treat the expression measurements as points in some metric space, where each observation (gene or sample, depending on the problem) contributes one point and the coordinates are given by the corresponding expression measures. Distances are computed in a pairwise manner within features (samples when genes are being compared). A different approach is to consider the data for each feature as an independent sample from a population. In this case, we are interested in questions such as whether the shape of the distribution of features is similar between two genes. For example whether they are bimodal or, perhaps have long right-tails. Other authors have also considered using distances between distributions as a means of analyzing genomic data. For example, (Butte and Kohane, 1999) suggest binning the data and then using a mutual information distance.

Alternatively, for each gene, across samples, we can consider the data as random I-vectors from some distribution. The simplest case is to assume that the expression measures for a particular gene follows an I-dimensional multivariate normal distribution with diagonal variance-covariance matrix.

Using, this approach, each gene provides a multivariate observation. Each of the I measurements for a given gene come from different samples, which are assumed to be independent, and hence the estimated variance-covariance matrix is diagonal. This approach can be used when both expression levels and their associated standard errors are available. The observed expression values are used to estimate the mean vector and the observed standard errors are used to estimate the variance-covariance matrix (Eisen et al., 1998).

Many different distance measures can be used to assess the similarities between two densities. We consider two measures that are not actually distances: the Kullback-Leibler information and Hamming's mutual information.

Definition 2.2.2
The Kullback-Leibler Information *(KLI) measure between densities f_1 and f_2 is defined as*

$$KLI(f_1, f_2) = E_1\{\log[\frac{f_1(X)}{f_2(X)}]\} \tag{2.7}$$

$$= \int \log[\frac{f_1(x)}{f_2(x)}] f_1(x)dx \tag{2.8}$$

where X is a random variable with density f_1 and E_1 denotes expectation with respect to f_1.

This ratio can be infinite and hence so can the KLI. The KLI is not a distance because it is not symmetric. KLI does not satisfy the triangle inequality either.

The KLI can be symmetrized in a number of ways, including the approach described in (Cook and Weisberg, 1982). They define the *Kullback-Leibler Distance*(KLD) to be,

$$2d_{KLD}(f_1, f_2) = KLI(f_1, f_2) + KLI(f_2, f_1) \tag{2.9}$$

The measure is symmetric and positive if f_1 and f_2 are different, however, it still does not satisfy the triangle inequality.

In the special case where $f_1 = N_m(\mu_1, \sum_1)$ and $f_2 = N_m(\mu_2, \sum_2)$ and assuming that $\sum_1$ and $\sum_2$ are positive definite, the expression for $d_{KLD}(f_1, f_2)$ simplifies and we get;

$$2d_{KLD}(f_1, f_2) = (\mu_1 - \mu_2)^T (\sum_2)^{-1}(\mu_1 - \mu_2) + \log(\frac{|\sum_1|}{|\sum_2|}) + \text{ tr } (\sum_1 (\sum_2)^{-1}) - m \tag{2.10}$$

However, this simplification involves making a strong assumption and requires knowledge of both variance-covariance matrices. Note that if $\sum_1$ and $\sum_2$ are identical, this is a form of Mahalanobis distance.

To compute between gene distances from the genomic data, the expression measures for a given gene, across samples, can be treated as a single observation from an I-dimensional multivariate normal distribution. For each gene, we estimate the mean in each coordinate (sample) by the observed expression measure for that sample, and we estimate the variances.

Closely related to the KLI is the *mutual information* (MI). The MI measures the extent to which two random variables X and Y are dependent. Let $f(\cdot, \cdot)$ denote the joint density function and $f_1(\cdot)$ and $f_2(\cdot)$ the two marginal densities for X and Y, respectively. Then the MI is defined as

Definition 2.2.3

$$MI(f_1, f_2) = E_f\Big\{ \log\Big[\frac{f(X,Y)}{f_1(X)f_2(Y)}\Big]\Big\} \tag{2.11}$$

and is zero in the case of independence. We note that like KLI, MI is not a distance although we will sometimes refer to it as if it were. This can easily

be determined by noticing the relationship between the MI distance and the KLI.

For our purpose, gene expression data on G genes for I genomic data samples may be summarized by a $G \times I$ matrix $X = (x_{gi})$, where x_{gi} denotes the expression measure of gene g in genomic data sample i. The expression levels might be either absolute or relative to the expression levels of a suitably defined common reference sample.

2.2.4 Distances and standardization

The behavior of the distance is closely related to the scale on which the observations have been made. Standardization of features is thus an important issue when considering distances between objects and is one method of making the features comparable. However, standardization also has the effect of removing some of the potentially interesting features in the data. Thus, in some cases it will be sensible to explore other approaches to obtaining comparability across features.

In the context of genomic data, one may standardize genes and/or samples. When standardizing genes, expression measures are transformed as follows

$$x_{gi} = \frac{x_{gi} - \text{ center } (x_{g.})}{\text{scale } (x_{g.})}$$

where center$(x_{g.})$ is some measure of the center of the distribution of the set of values $x_{gi}, i = 1, \cdots, I$, such as mean or median, and scale $(x_{g.})$ is a measure of scale such as the standard deviation, interquartile range, or MAD (median absolute deviation about the median).

Alternatively, one may want to standardize samples if there is interest in clustering or classifying them (rather than clustering or classifying the genes). Now we use

$$x_{gi} = \frac{x_{gi} - \text{ center } (x_{.i.})}{\text{scale } (x_{.i})}$$

where the centering and scaling operations are carried out across all genes measured on sample i.

We now consider the implications of the preceding discussion on standardization in the context of both relative mRNA expression measurements and absolute mRNA expression measurements.

Consider the standard situation where x_{gi} represetns the expression measure on a log scale for gene g on patient (i.e., array or sample) i. Let $y_{gi} = x_{gi} - x_{gA}$ where patient A is our reference. Then, the relative expression measures y_{gi} correspond to the standard data available from a cDNA experiment with a common reference. The use of relative expression measures represents a location transformation for each gene (gene centering). Now, suppose that

we want to measure the distance between patient samples i and j.Then, for the classes of distances considered in Equation 2.1

$$d(\mathbf{y}_{.i}, \mathbf{y}_{.j}) = \sum_{g=1}^{G} d_g(y_{gi}, y_{gj}) = \sum_{g=1}^{G} d_g(x_{gi} - x_{gA}, x_{gj} - x_{gA})$$

When the $d_g(x, y)$ are functions of $x - y$ alone, then $d(\mathbf{y}_{.i}, \mathbf{y}_{.j}) = d(\mathbf{y}_{.i}, \mathbf{y}_{.j})$ and it does not matter if we look at relative (the $\mathbf{y}$'s) or absolute (the $\mathbf{x}$'s) expression measures.

Suppose that we are interested instead in comparing genes and not samples. Then the distance between genes g and h is

$$d(\mathbf{y}_{g.}, \mathbf{y}_{h.}) = \sum_{i=1}^{I} d_i(y_{gi}, y_{hi}) = \sum_{i=1}^{I} d_i(x_{gi} - x_{gA}, x_{hi} - x_{hA})$$

If $d(\mathbf{x}, \mathbf{y})$ has the property that $d(\mathbf{x} - \mathbf{c}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})$ for any c, then the distance measure is the same for absolute and relative expression measures.

Chapter 3

Work

3.1 Introduction of Simulation

Lets say that we are interested in predicting the run time of an athlete depending on his shoe size, height and weight in a study of 100 people. We can do so using a simple multiple linear regression model where

$$y = \beta_0 + \beta_1 * height + \beta_2 * weight + \beta_3 * shoesize$$

Here y is the response variable (run time), n is the number of observations (100 people), p is the number of variables/ features/ predictors, X is a $n \times p$ matrix.

This data set is a low dimensional data where $n >> p$ but most of the biological data sets coming out of modern biological techniques are high dimensional of $n << p$. This poses statistical challenge and simple linear regression can no longer help us.
For example,

- Identify the risk factors(genes) for prostrate cancer based on gene expression data
- Predict the chances of breast cancer survival in a patient.
- Identify patterns of gene expression among different sub types of breast cancer

In all of the 3 examples, listed above n, number of observations, is 30-40 patients whereas p, number of features, is approximately 30,000 genes. Listed below are things that can go wrong with high dimensional data - some of these predictors are useful, some are not - if we include too many predictors, we can over fit the data. This is why we need Machine Learning and here is breif explanation of machine learning terms that we use frequently in this thesis and related literature review. More detail information can be found in the book of (Friedman, Hastie, and Tibshirani, 2001).

- **Supervised Learning**: Use a data set X to predict the association with a response variable Y. The response variable can be continuous or categorical. For example: Predicting the chances of breast cancer survival in a patient.

- **Unsupervised Learning**: Discover the associations or patterns in X. No response variable is present. For example: Cluster similar genes into groups.
- **Training & Test Datasets**: Usually we split observation into test and training data sets. We fit the model on the training data set and evaluate on the test data set. The test set error rate is an estimate of the models performance on future data sets.
- **Model Selection**: We usually consider numerous models for a given problem. For example, we are trying to identify the genes responsible for a given disease using gene expression data set- we could have the following models
    1. **Model 1**: Use all 30000 genes from the array to build a model
    2. **Model 2**: We include only genes related to the pathway that we know is upregulated in that disease to build a model
    3. **Model 3** - Include genes found in literature which are known to influence this disease It is highly recommended to use the test set only on our final model to see how our model will do with new, unseen data. So how do we pick the best model which can be tested on the test data set?

To discuss about which model is going to use, we need further concepts as shown below.

- **Cross-validation**: We can use different approaches to find the best model. Lets look at the commonly used approaches, namely, validation set, leave one out cross-validation, *k*-fold cross validation.
- **Validation set approach**: It deals with diving the full data sets into 3 groups - training set, validation set and the test set. We train the models on the training set, evaluate their performance on the validation set and then the best model is chosen to fit on the test set.
- **Leave one out cross validation**: It starts with fitting n models (where n is number of observations in the training data set), each on $n-1$ observations, evaluating each model on the left-out observation. The best model is the one for which the total test error is the smallest and that is then used to predict the test set.
- **5 fold cross validation** (here k=5): It is splitting the training data set into 5 sets and repeatedly training the model on the other 4 sets and evaluating the performance on the fifth.
- **Bias, Variance, Overfitting**: Bias refers to the average difference between the actual betas and the predicted betas, Variance refers to the amount by which the betas differ across experiments. As the model complexity(no of variables) increases, the bias decreases and the variance increases. This is know as the Bias-Variance Tradeoff and a model that has too much of variance, is said to be over fit.

```
## DataFrame with 8 rows and 9 columns
##           SampleName     cell      dex    albut       Run avgLength
##             <factor> <factor> <factor> <factor>  <factor> <integer>
## SRR1039508 GSM1275862   N61311    untrt    untrt SRR1039508       126
## SRR1039509 GSM1275863   N61311     trt     untrt SRR1039509       126
## SRR1039512 GSM1275866   N052611   untrt    untrt SRR1039512       126
## SRR1039513 GSM1275867   N052611    trt     untrt SRR1039513        87
## SRR1039516 GSM1275870   N080611   untrt    untrt SRR1039516       120
## SRR1039517 GSM1275871   N080611    trt     untrt SRR1039517       126
## SRR1039520 GSM1275874   N061011   untrt    untrt SRR1039520       101
## SRR1039521 GSM1275875   N061011    trt     untrt SRR1039521        98
##           Experiment    Sample    BioSample
##             <factor>  <factor>     <factor>
## SRR1039508  SRX384345 SRS508568 SAMN02422669
## SRR1039509  SRX384346 SRS508567 SAMN02422675
## SRR1039512  SRX384349 SRS508571 SAMN02422678
## SRR1039513  SRX384350 SRS508572 SAMN02422670
## SRR1039516  SRX384353 SRS508575 SAMN02422682
## SRR1039517  SRX384354 SRS508576 SAMN02422673
## SRR1039520  SRX384357 SRS508579 SAMN02422683
## SRR1039521  SRX384358 SRS508580 SAMN02422677
```

FIGURE 3.1: Descriptive Statistics of Datasets, `airway` from (Himes et al., 2014)

3.1.1 Data Set

For **Unsupervised Learning**, we will use RNA-Seq count data from the Biocoductor package, `airway`. (Himes et al., 2014) From the abstract, a brief description of the RNA-Seq experiment on airway smooth muscle (ASM) cell lines: "Using RNA-Seq, a high-throughput sequencing method, we characterized transcriptomic changes in four primary human ASM cell lines that were treated with dexamethasone - a potent synthetic glucocorticoid (1 micromolar for 18 hours)."

For **Supervised Learning**, we will use cervical count data from the Biocoductor package, `MLSeq`. (Goksuluk et al., 2019) This data set contains expressions of 714 miRNA's of human samples. There are 29 tumor and 29 non-tumor cervical samples. For learning purposes, we can treat these as two separate groups and run various classification algorithms.

3.2 Unsupervised Learning

Unsupervised Learning is a set of statistical tools intended for the setting in which we have only a set of 'p' features measured on 'n' observations. We are primarily interested in discovering interesting things about the 'p' features.

Unsupervised Learning is often performed as a part of Exploratory Data Analysis. These tools help us to get a good idea about the data set. Unlike a supervised learning problem, where we can use prediction to gain some confidence about our learning algorithm, there is no way to check our model.

3.2.1 RLOG TRANSFORMATION

Many common statistical methods for exploratory analysis of multidimensional data, especially methods for clustering and ordination (e.g., principal-component analysis and the like), work best for (at least approximately) homoskedastic data; this means that the variance of an observed quantity (here, the expression strength of a gene) does not depend on the mean.

In RNA-Seq data, the variance grows with the mean.If one performs PCA (principal components analysis) directly on a matrix of normalized read counts, the result typically depends only on the few most strongly expressed genes because they show the largest absolute differences between samples.

To assess overall similarity between samples: Which samples are similar to each other, which are different? Does this fit to the expectation from the experiment's design? We use the R function dist to calculate the Euclidean distance between samples. To avoid that the distance measure is dominated by a few highly variable genes, and have a roughly equal contribution from all genes, we use it on the rlog-transformed data.

3.2.2 HEATMAP

We visualize the sample-to-sample distances in a heatmap, using the function `heatmap.2` from the `gplots` package. Note that we have changed the row names of the distance matrix to contain treatment type and patient number instead of sample ID, so that we have all this information in view when looking at the heatmap.

3.2.3 PCA

Another way to visualize sample-to-sample distances is a principal-components analysis (PCA). In this method, the data points (i.e., here, the samples) are projected onto the 2D plane such that they spread out in the two directions which explain most of the differences in the data. The x-axis is the direction (or principal component) which separates the data points the most. The amount of the total variance which is contained in the direction is printed in the axis label. Here, we have used the function `plotPCA` which comes with

```
##           SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## SRR1039509   40.89060
## SRR1039512   37.35231   50.07638
## SRR1039513   55.74569   41.49280   43.61052
## SRR1039516   41.98797   53.58929   40.99513   57.10447
## SRR1039517   57.69438   47.59326   53.52310   46.13742   42.10583
## SRR1039520   37.06633   51.80994   34.86653   52.54968   43.21786
## SRR1039521   56.04254   41.46514   51.90045   34.82975   58.40428
##           SRR1039517 SRR1039520
## SRR1039509
## SRR1039512
## SRR1039513
## SRR1039516
## SRR1039517
## SRR1039520   57.13688
## SRR1039521   47.90244   44.78171
```

FIGURE 3.2: Euclidean distance between samples, `MLSeq` from (Goksuluk et al., 2019)

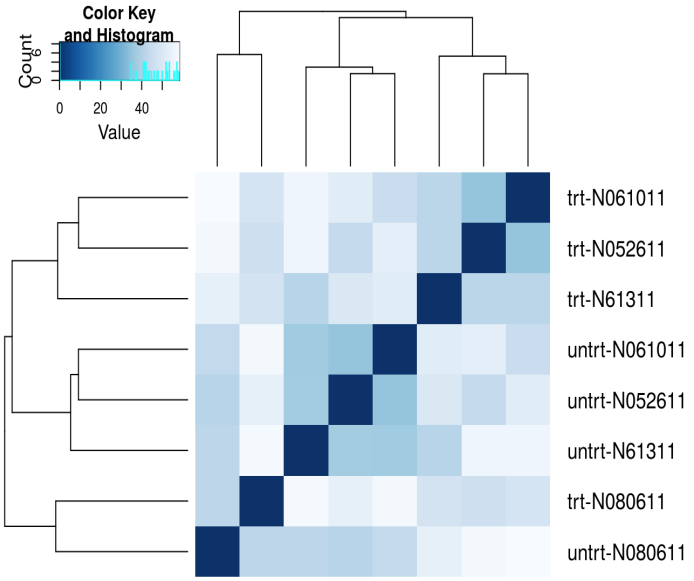

FIGURE 3.3: HEATMAP, `MLSeq` from (Goksuluk et al., 2019)

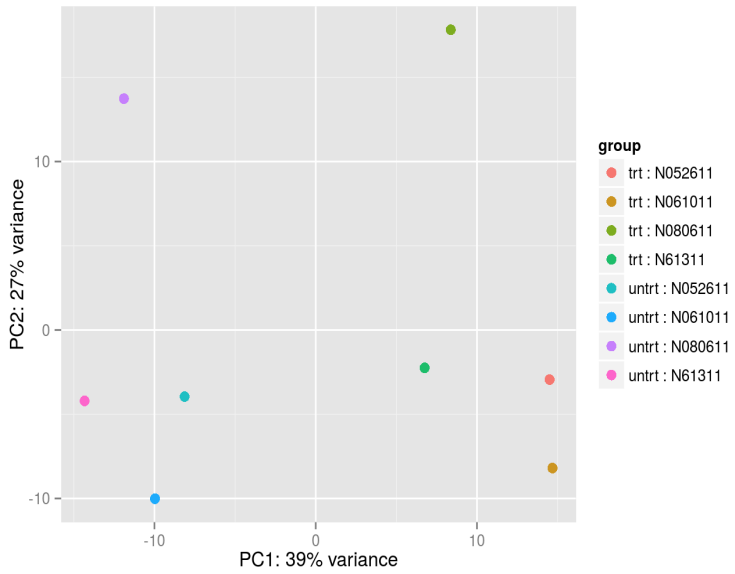

FIGURE 3.4: PCA, `MLSeq` from (Goksuluk et al., 2019)

`DESeq2`. The two terms specified by intgroup are the interesting groups for labelling the samples; they tell the function to use them to choose colors.

From Figure 3.3 and Figure 3.4 visualizations, we see that the differences between cells are considerable, though not stronger than the differences due to treatment with dexamethasone. This shows why it will be important to account for this in differential testing by using a paired design ("paired", because each dex treated sample is paired with one untreated sample from the same cell line). We are already set up for this by using the design formula `cell + dex` when setting up the data object in the beginning.

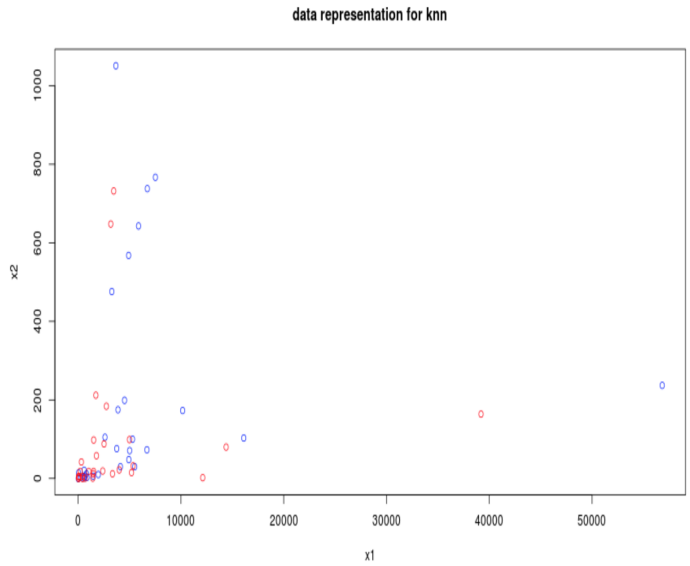

FIGURE 3.5: Data Representation for KNN

3.3 Supervised Learning

In supervised learning, along with the features $X_1, X_2, \cdots, X_p$, we also have the a response Y measured on the same n observations. The goal is then to predict Y using $X_1, X_2, \cdots, X_p$ for new observations.

3.3.1 K-Nearest Neighbors (KNN)

For the cervical data, we know that the first 29 are non-Tumor samples whereas the last 29 are Tumor samples. We will code these as 0 and 1 respectively.

Let's look at one of the most basic supervised learning techniques k-Nearest Neighbor and see what all goes into building a simple model with it. For the sake of simplicity, we will use only 2 predictors (so that we can represent the data in 2 dimensional space).

Given a observation x_0 and a positive integer, K, the KNN classifier first identifies K points in the training data that are closest to x_0, represented by N_0. It estimates the conditional probability for class j as a fraction of N_0 and applies Bayes rule to classify the test observation to the class with the largest probability. More concretely, if $k = 3$ and there are 2 observation belonging to class 1 and 1 observation belonging to class 2, then we the test observation is assigned to class1.

For all supervised experiments its a good idea to hold out some data as Training Data and build a model with this data. We can then test the built model using the left over data (Test Data) to gain confidence in our model. We will randomly sample 30 % of our data and use that as a test set. The remaining 70 % of the data will be used as training data.

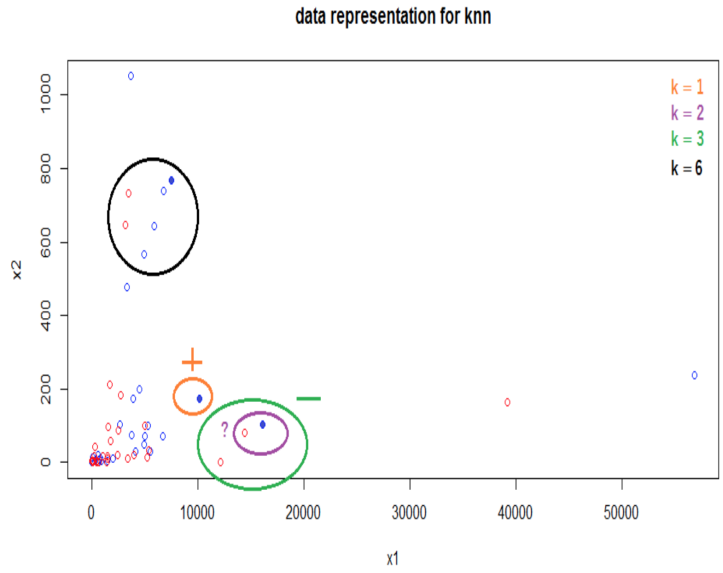

FIGURE 3.6: Data Representation for KNN

Training set error is the proportion of mistakes made if we apply our model to the training data and Test set error is the proportion of mistakes made when we apply our model to test data. For different neighbors, let us calculate the training error and test error using KNN.

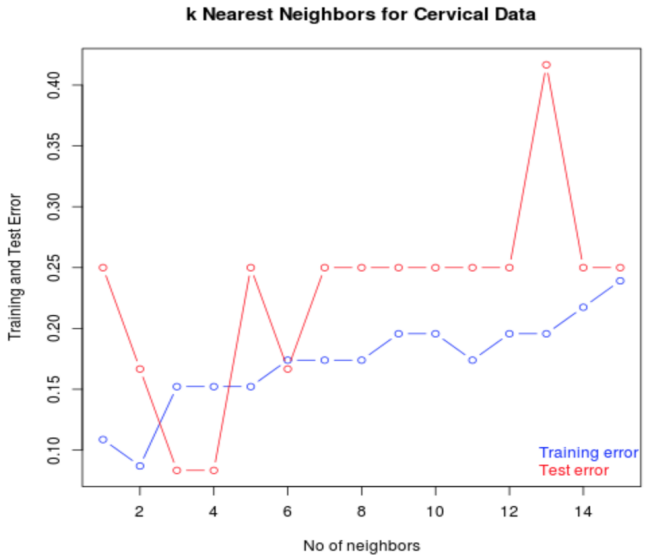

FIGURE 3.7: KNN for Cervival Data

Chapter 4

Discussion

4.1 Challenges

In bioinformatics, much research has already applied machine learning techniques on genomic data and human genome (Larranaga et al., 2006) and (Plewczynski et al., 2006). Common to most of the projects, are the focus on optimizing performance within a single or specific context. A common challenge which often arises, is the challenges of *imbalanced data*.

4.1.1 Imbalanced data

The imbalanced data problem (He and Garcia, 2008) is the challenge of learning from data where there are more samples of a given class (or concept) than others. The imbalance between two or more classes are called *between-class* imbalance, while imbalance inside a given class is called *within-class* imbalance. The class with most examples are denoted the *majority class*, while a class with relatively few samples are denoted the *minority class*. The distinction between majority and minority classes are usually done when the imbalance reaches a certain (imbalance) *ratio*, e.g. 1:2, 1:10, 1:100, 1:1000 or more.

4.2 Conclusion

Distances are an integral part of all machine learning algorithms and hence play a central role in the analysis of most genomic data. The distance that is used for any particular task can have a profound effect on the output of the machine learning method and thus, it is essential that users ensure that the same distance method is used when comparing machine learning algorithms.

In this thesis, we implemented the supervised and unsupervised machine learning algorithms and looked at the importance of the notion of distance in the genomic data.

4.3 Future Work

Eventually, we would like to connect the genomic data with *Precision Medicine* in the future study. *Precision Medicine* seeks to maximize the quality of health care by individualizing the health-care process to the uniquely evolving health status of each patient. This work will expand a broad range of scientific areas including drug discovery, genetics/genomics, health communication, and causal inference, all in support of evidence-based, i.e., data-driven, decision making.

Appendix A

Appendix

A.1 R Code

```
library(airway)
data("airway")
se <- airway
colData(se)

library("DESeq2")
dds <- DESeqDataSet(se, design = ~ cell + dex)

library(MLSeq)
filepath = system.file("extdata/cervical.txt", package = "MLSeq")
cervical = read.table(filepath, header = TRUE)

rld <- rlog(dds)
head(assay(rld))

sampleDists <- dist( t( assay(rld) ) )
sampleDists

library("gplots")
library("RColorBrewer")

# HEATMAP
sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- paste( rld$dex, rld$cell, sep="-" )
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
hc <- hclust(sampleDists)
heatmap.2( sampleDistMatrix, Rowv=as.dendrogram(hc),
          symm=TRUE, trace="none", col=colors,
          margins=c(2,10), labCol=FALSE )

# PCA Plot
plotPCA(rld, intgroup = c("dex", "cell"))


# Supervised Learning of KNN

class = data.frame(condition = factor(rep(c(0, 1), c(29, 29))))
data <- t(cervical)
data <- data[,1:2]
df   <- cbind(data, class)
colnames(df) <- c("x1","x2","y")
rownames(df) <- NULL
head(df)

plot(df[,"x1"], df[,"x2"], xlab="x1", ylab="x2",
     main="Data Representation for k-nearest neighbors",
     col=ifelse(as.character(df[,"y"])==1, "red","blue"))


# Tranining and Test Error
set.seed(9)
nTest = ceiling(ncol(cervical) * 0.2)
```

```
ind = sample(ncol(cervical), nTest, FALSE)

cervical.train = cervical[, -ind]
cervical.train = as.matrix(cervical.train + 1)
classtr = data.frame(condition = class[-ind, ])

cervical.test = cervical[, ind]
cervical.test = as.matrix(cervical.test + 1)
classts = data.frame(condition = class[ind, ])

library(class)

newknn <- function( testset, trainset, testclass, trainclass, k)
{
    pred.train <- knn.cv(trainset, trainclass, k=k)
    pred.test <- knn(trainset, testset, trainclass, k=k)

    test_fit <- length(which(mapply(identical, as.character(pred.test),
        testclass)==FALSE))/length(testclass)

    train_fit <- length(which(mapply(identical, as.character(pred.train),
            trainclass)==FALSE))/length(trainclass)

    c(train_fit=train_fit, test_fit= test_fit)
}

trainset <- t(cervical.train)
testset <- t(cervical.test)
testclass <- t(classts)
trainclass <- t(classtr)
klist <- 1:15
ans <- lapply(klist, function(x)
    newknn(testset, trainset, testclass, trainclass,k =x))

resdf <- t(as.data.frame(ans))
rownames(resdf) <- NULL
plot(klist, resdf[,"train_fit"], col="blue", type="b",ylim=c(range(resdf)),
    main="k Nearest Neighbors for Cervical Data", xlab="No of neighbors",
    ylab ="Training and Test Error")
points(klist, resdf[,"test_fit"], col="red", type="b")
legend("bottomright", legend=c("Training error","Test error"),
    text.col=c("blue","red"), bty="n")
```

Bibliography

Alpaydin, Ethem (2009). *Introduction to machine learning*. MIT press.

Altland, Henry W (1999). *Regression analysis: statistical modeling of a response variable*.

Butte, Atul J and Isaac S Kohane (1999). "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements". In: *Biocomputing 2000*. World Scientific, pp. 418–429.

Consortium, 1000 Genomes Project et al. (2012). "An integrated map of genetic variation from 1,092 human genomes". In: *Nature* 491.7422, p. 56.

Cook, R Dennis and Sanford Weisberg (1982). *Residuals and influence in regression*. New York: Chapman and Hall.

Duda, Richard O, Peter E Hart, and David G Stork (2012). *Pattern classification*. John Wiley & Sons.

Eisen, Michael B et al. (1998). "Cluster analysis and display of genome-wide expression patterns". In: *Proceedings of the National Academy of Sciences* 95.25, pp. 14863–14868.

Eisenstein, Michael (2015). "The power of petabytes". In: *Nature* 527.7576, S2.

França, Lilian TC, Emanuel Carrilho, and Tarso BL Kist (2002). "A review of DNA sequencing techniques". In: *Quarterly reviews of biophysics* 35.2, pp. 169–200.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York.

Gentleman, Robert et al. (2006). *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer Science & Business Media.

Goksuluk, Dincer et al. (2019). "MLSeq: Machine Learning Interface to RNA-Seq Data". In:

Harrow, Jennifer et al. (2012). "GENCODE: the reference human genome annotation for The ENCODE Project". In: *Genome research* 22.9, pp. 1760–1774.

He, Haibo and Edwardo A Garcia (2008). "Learning from imbalanced data". In: *IEEE Transactions on Knowledge & Data Engineering* 9, pp. 1263–1284.

Himes, Blanca E et al. (2014). "RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells". In: *PloS one* 9.6, e99625.

Kent, W James et al. (2010). "BigWig and BigBed: enabling browsing of large distributed datasets". In: *Bioinformatics* 26.17, pp. 2204–2207.

Kircher, Martin (2012). "Analysis of high-throughput ancient DNA sequencing data". In: *Ancient DNA*. Springer, pp. 197–228.

Larranaga, Pedro et al. (2006). "Machine learning in bioinformatics". In: *Briefings in bioinformatics* 7.1, pp. 86–112.

Moore, Gordon E et al. (1965). *Cramming more components onto integrated circuits*.

Penny, Kay I (1996). "Appropriate critical values when testing for a single multivariate outlier by using the Mahalanobis distance". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 45.1, pp. 73–81.

Plewczynski, Dariusz et al. (2006). "Support-vector-machine classification of linear functional motifs in proteins". In: *Journal of molecular modeling* 12.4, pp. 453–461.

Quinlan, Aaron R and Ira M Hall (2010). "BEDTools: a flexible suite of utilities for comparing genomic features". In: *Bioinformatics* 26.6, pp. 841–842.

Schuster, Stephan C (2007). "Next-generation sequencing transforms today's biology". In: *Nature methods* 5.1, p. 16.

Tomczak, Katarzyna, Patrycja Czerwińska, and Maciej Wiznerowicz (2015). "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemporary oncology* 19.1A, A68.

Wetterstrand, Kris A (2013). *DNA sequencing costs: data from the NHGRI Genome Sequencing Program (GSP)*.